

ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

А.Н. СУВОРОВ (Институт физики имени Л.В. Киренского СО РАН)



Трудно назвать сферу деятельности человека, где в настоящий момент не используются элементы автоматизации. Решения в основном основаны на применении микропроцессоров. Массовое применение микропроцессоров, в свою очередь стимулирует развитие или эволюцию элементной базы. Наибольшую часть рынка средств комплексной автоматизации занимает промышленное производство.

“Верхние” уровни автоматизации достаточно хорошо развиты, это задачи автоматизации управления производством (АУП). Наиболее узким местом являются автоматизированные системы управления технологическими процессами (АСУ ТП), и как часть этих систем программируемые логические контроллеры (ПЛК). Основным стандартом для разработки систем управления на основе ПЛК является стандарт IEC 61131-3. Системы, построенные на основе данного стандарта имеют ряд недостатков:

- не всегда обеспечивают требуемый уровень надежности;
- сложность и как следствие высокая стоимость реализации построения распределенных систем;
- распределенные системы строятся по иерархическому принципу, что является дополнительным “слабым звеном”;
- отсутствие санкционирования доступа (если вирус попал в систему, то вся система будет заражена. Очень хорошо данный вопрос рассмотрен в статье [1]);
- невозможность в полном объеме использовать вычислительные и управляющие ресурсы ПЛК, сконструированных с использованием современной элементной базы.

Для решения данных недостатков была разработана методология, представленная в данной статье.

Ключевые слова: агентно-ориентированные информационно-управляющие системы (РИУС), стандарт IEC 61499, стандарт IEC 61131-3, языки программирования ISO/IEC 14882:1998 (“C”), ISO/IEC 14882:2003 (“C++”), IEC 61131-3 (“FDB, ST”), Posix OC NUTTX, Linux.

ЦЕЛИ И ЗАДАЧИ ИССЛЕДОВАНИЯ

Целью работы является разработка методологии построения распределенных, агентно-ориентированных информационно-управляющих систем (РИУС). Для достижения поставленной цели необходимо решить следующую основную задачу: разработать методологию проектирования РИУС управления промышленными процессами на основе стандарта IEC 61131-3 и агентно-ориентированного подхода в программировании.

ПРЕДМЕТ ИССЛЕДОВАНИЯ

Предметом исследования являются методы и средства проектирования и организа-

ции программного кода, для решения задач кросс-платформенности, масштабируемости и реализации сложных взаимодействий между узлами РИУС.

МЕТОДЫ ИССЛЕДОВАНИЯ

Максимально использовать существующие стандарты вкупе с агентно-ориентированным подходом в программировании.

НАУЧНАЯ НОВИЗНА

Научная новизна статьи определяется следующими результатами.

Попытка устранить проблемы, возникающие при использовании существующих стандартов/методологий.

СТАНДАРТ IEC 61499

Для построения распределенных систем, лишенных перечисленных выше недостатков, был принят стандарт IEC 61499. Стандарт основан на IEC 61131-3, до сих пор находится в разработке и не позволяет в полном объеме избавиться от недостатков IEC 61131-3. Принятие этого стандарта затягивалось долгими перипетиями. Основная причина — нечеткое предъявление требований к стандарту специалистами по разработке управляющих систем.

В основе стандарта лежит понятие функциональных блоков [2]. Введение этого понятия — по сути, первый шаг на принятии агентно-ориентированного подхода, как элемента стандарта при создании РИУС.

ПРЕДЛАГАЕМАЯ МЕТОДОЛОГИЯ

Предлагаемая методология в основе также содержит стандарт IEC 61131-3, расширяя его понятиями агентно-ориентированного подхода в программировании.

При создании РИУС предлагается использовать как элементы системы свободно программируемые ПЛК. В качестве языков программирования могут быть использованы языки программирования ISO/IEC 14882:1998 ("C"), ISO/IEC 14882:2003 ("C++"), IEC 61131-3 ("FDB, ST"), причем все они могут быть использованы как интерпретаторы.

ПОНЯТИЕ "ЭКЗЕМПЛЯР"

Рассмотрим понятие "instance" (экземпляр). В самом стандарте определение этого понятия не приводится, хотя неоднократно

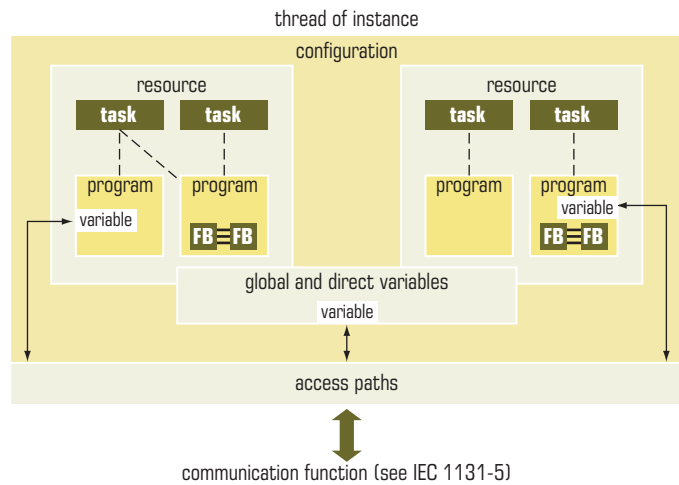


Рис. 1. Диаграмма конфигурации элементов IEC-61131-3

используется. Применяется для описания локализации места, где физически будет функционировать конфигурация [3]. Саму конфигурацию предлагается рассматривать как поток (процесс), работающий под управлением операционной системы (рис. 1).

ПРИМЕНЕНИЕ ОПЕРАЦИОННЫХ СИСТЕМ

Ситуацию с неэффективным использованием возможности оборудования отчасти можно исправить применением Операционных систем — ОС. На ПЛК гарвардской архитектуры это может быть Posix ОС NUTTX, на архитектуре фон Неймана — Linux.

Если рассматривать архитектуру экземпляра, где присутствует ОС (например Linux) в качестве системы распределения и управления ресурсами, то напрашивается следующее дополнение стандарта IEC-61131-3, рис. 2:

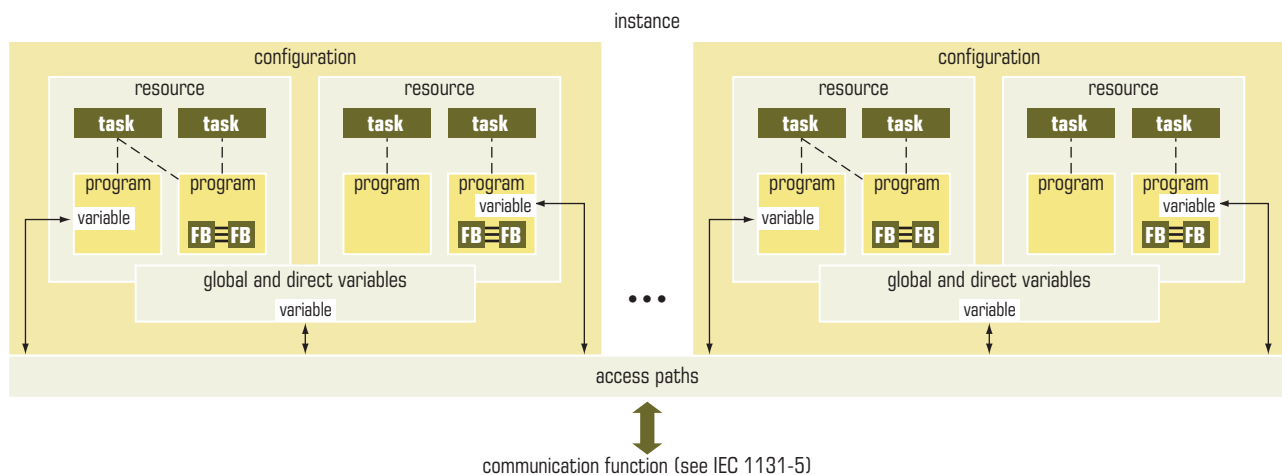


Рис. 2. Диаграмма конфигурации элементов для ОС

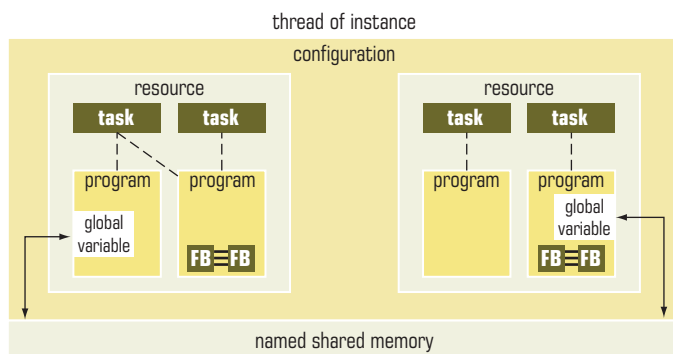


Рис. 3. Поток-конфигурация

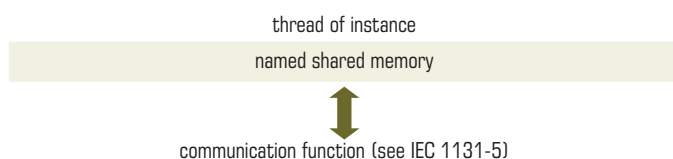


Рис. 4. Поток обслуживания коммуникационных функций

То есть, для более рационального использования ресурсов РИУС, разумно, когда на экземпляре будут функционировать несколько конфигураций.

Глобальные переменные (сигналы) предлагается реализовать в виде именованной разделяемой памяти. Таким образом, мы обеспечиваем атомарность транзакций и доступ из любых процессов/потоков экземпляра.

Пути доступа к переменным представлены на рис. 3, 4.

На контроллерах без операционной системы в роли named share memory выступает секция глобальных переменных.

Структура блока именованной разделяемой памяти организована в виде таблицы 1 и выглядит следующим образом:

Таблица 1. Структура блока именованной разделяемой памяти

Name	Value	Access	Update	Sync
char[36]	int64	char[1]	int64	int64
Data				
Data				
Data				
...				
Data				

Name; // Name of signal/variable
 Value; // Value
 Access; // Access R/W
 Update; // Last time of update
 Sync; // Last time of synchronyze

АГЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД

Существует несколько толкований смысла парадигмы “Агентно-ориентированный подход” (АОП). Наиболее широко используются идеи, предложенные Шохемом [4].

Шохем, представляя агентно-ориентированный подход как новую парадигму, достаточно подробно обсуждает, в каком смысле в ней используется слово агент.

В нашем случае это “умный” программный модуль, решающий определенный класс задач. Агент может содержать в себе от одного до N независимо выполняющихся потоков. Каждый поток может выполняться с разным приоритетом и разным временем выполнения цикла работы. Например, поток работы с датчиками температуры – опрос раз в 5 секунд. Поток работы с датчиками давления – частота опроса 1 кГц.

По сути, агент включает в себя объекты. Объектами могут быть блоки конфигурации, блоки принятия решений, блоки работы с датчиками/исполнительными механизмами.

Схематически это выглядит следующим образом (рис. 5):

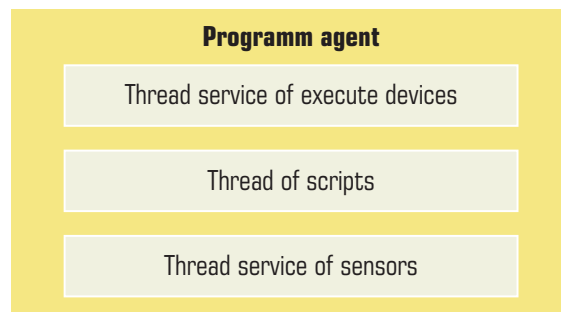


Рис. 5. Схема агента

Объекты синхронизируют свое состояние с разделяемой памятью (рис. 6).

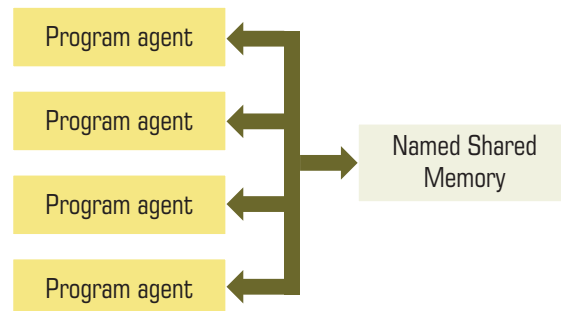


Рис. 6. Схема синхронизации данных агентов с разделяемой памятью

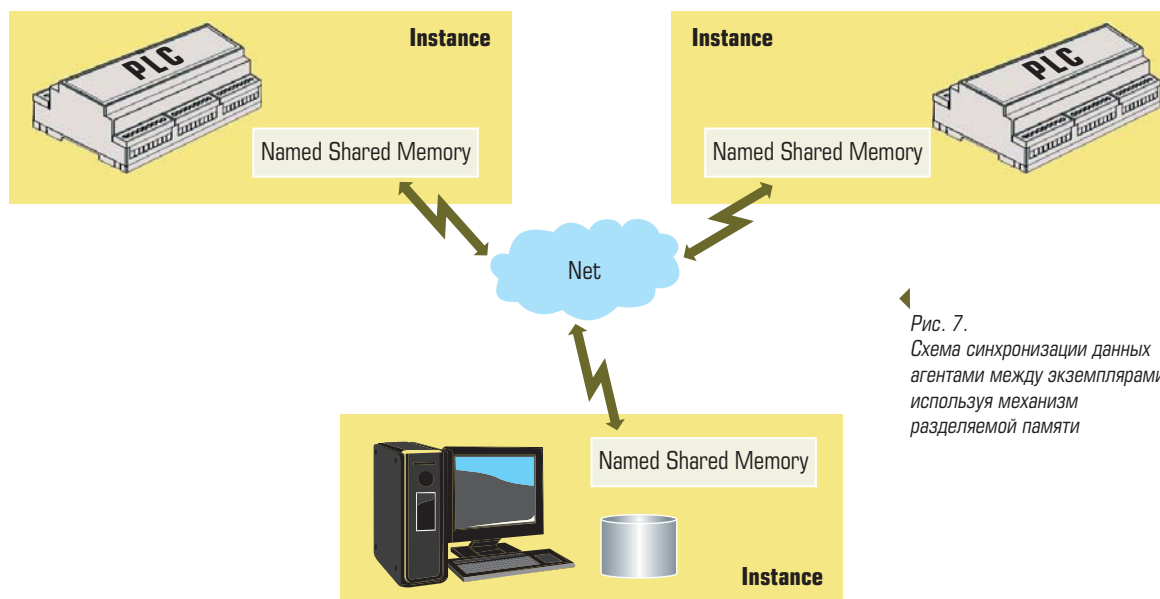


Рис. 7.
Схема синхронизации данных
агентами между экземплярами,
используя механизм
разделяемой памяти

Также агенты отвечают за построение и работу сети экземпляров, синхронизируя строки разделяемой памяти между узлами (экземплярами) системы.

Схематически это представлено на рис. 7.

Агенты могут работать в любой среде (ОС Windows, Linux, ...). Агенты принимают решения работать автономно или в составе сети.

Например, схема взаимодействия агентов по обмену данными может быть представлена в виде нерегулярного графа (рис. 8).

$$G := (V, E),$$

где G – неориентированный граф,

V – это непустое множество вершин или узлов (экземпляров) РИУС,

E – множество пар вершин, называемых рёбрами, или формализованные физические каналы передачи данных.

Агент анализирует текущее состояние каналов и по определенным параметрам (весам) принимает решения о формировании маршрутов передачи данных. В случае нарушения всех маршрутов к данному экземпляру, принимается решение работать автономно.

ЗАКЛЮЧЕНИЕ

Используя предложенную методологию уже созданы и находятся в промышленной эксплуатации реализации РИУС.

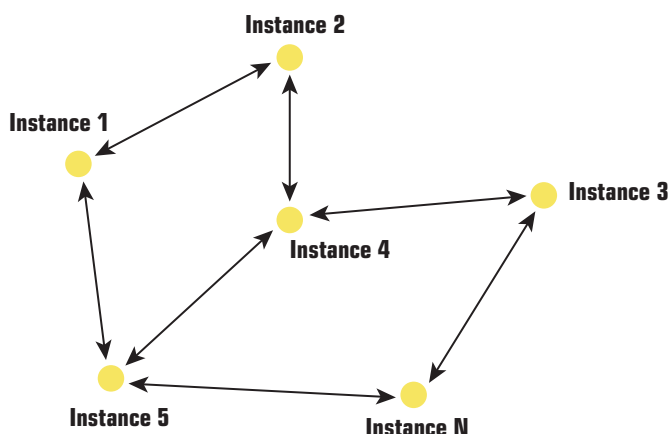


Рис. 8. Схематическое изображение маршрутов обмена данными

Список литературы

1. *Вирс, живущий исключительно в ПЛК* [Электронный ресурс] URL <https://habrahabr.ru/post/302276/> 31.05.2016.
2. *Гулько С.В. Обзор стандарта IEC 61499.* Гулько С.В. "ХОЛИТ Дэйта Системс", г. Киев. Николас Джоврей, ICP Triplex ISaGRAF, Франция. С. 9-12.
3. *The third edition of IEC 61131-3 is approved as International Standard.* С. 15.
4. *Yoav Shoham Agent-oriented programming* Robotics Laboratory Computer Science Department, Stanford University Stanford, CA 94305, USA. С. 52-54.

Суворов Александр Николаевич – ведущий инженер лаборатории молекулярной спектроскопии, Институт физики имени Л.В. Киренского СО РАН.